

REMARKS

In response to the Final Office Action mailed November 14, 2008, and Advisory Action mailed February 25, 2009, Applicant respectfully requests reconsideration and entry of this amendment. Claims 1-5, 7-12, 14-19, 21-26, and 28 were previously pending in this application. By this amendment, Applicant is canceling claims 3, 10, 17 and 24 without prejudice or disclaimer. Claims 1-5, 7-12, 14-19, 21-26, and 28 have been amended. As a result, claims 1, 2, 4, 5, 7-9, 11, 12, 14-16, 18, 19, 21-23, 25, 26, and 28 are pending for examination with claims 1, 8, 15, and 22 being independent. No new matter has been added.

Rejections Under 35 U.S.C. §103 based on AAPA, Craft, Oesterreicher and Boucher

The Office Action rejected claims 1, 5, 7, 15, 19, and 21 (including independent claims 1 and 15) under 35 U.S.C. §103(a) as allegedly being unpatentable over Applicant's Admitted Prior Art ("AAPA") in view of Craft et al., US Patent No. 6,687,758 ("Craft"), in view of Oesterreicher et al., US 2004/0006636 ("Oesterreicher"), and in further view of Boucher et al., 2004/0240435 ("Boucher"). Applicant respectfully disagrees.

A. Independent Claim 1

Claim 1, as amended, recites:

A method for transferring control between a first network interface controller and at least a second network interface controller in a multiple network interface device, the method comprising:

after the first network interface controller sends an identifier associated with a memory location in the multiple network interface device to a second device and the identifier and an associated data field are subsequently received by the second network interface controller in the multiple network interface device from the second device, receiving, by a program component of the multiple network interface device, the identifier associated with the memory location in the multiple network interface device and the associated data field from the second device, wherein the second network interface controller has no knowledge of the identifier and the associated data field, and wherein the first network interface controller and the second network interface controller operate under a Remote Direct Memory Access (RDMA) protocol;

querying the first network interface controller to supply the program component with a list of valid identifiers generated by the first network interface

controller, wherein each identifier from the list of valid identifiers is associated with a location in a memory of the multiple network interface device;

determining whether the first network interface controller generated the identifier, wherein when the first network controller generated the identifier the list of valid identifiers comprises the identifier;

when it is determined that the first network interface controller generated the identifier, transmitting the memory location associated with the identifier to the second network interface controller, wherein the second network interface controller subsequently transmits the associated data field to the memory location; and

when the identifier is not found among the list of valid identifiers, invalidating the identifier and discarding the associated data field.

On page 5, the Office Action states that the previously issued Office Action “was clear in referring to the packet as the message indicating the reception of the identifier. The Office action also referred to the packet as inherently containing the identifier in the header section of the packet. The fact that Craft does not explicitly discuss having an identifier in the header section of the packet, and just broadly discusses generating the packet summary and comparing a hash of the packet summary with the hash table corresponding to the CCB, should not be taken as an indication that such identifier does not exist in the packet of Craft.” Applicant respectfully notes that claim 1 recites an identifier *associated with a memory location* in the multiple network interface device (emphasis added). Furthermore, claim 1 has been amended to recite, inter alia, receiving, by a program component of the multiple network interface device, *the identifier associated with the memory location* in the multiple network interface device and the associated data field (emphasis added). Applicant does not argue that Craft does not discuss **any** identifier. Rather, Craft does not teach or suggest receiving ... the identifier **associated with the memory location**, as recited in claim 1 (emphasis added).

On page 4, the Office Action states, referring to Applicant’s previously filed response, “Applicant argues that: “the packet described in Craft is different from an identifier associated with a memory location in the multiple network interface device”. This argument is not persuasive because the claim language is lacking specificity as to how the claimed “identifier associated with the memory location” is structurally and functionally different from an identifier inherently contained in the packet summary of Craft.” Applicant respectfully notes that Craft does not describe such identifier and the packet of Craft therefore simply **does not** comprise such identifier. Indeed, since Craft does not discuss the RDMA protocol, as agreed to on page 3 of

the Office Action, Craft does not teach the identifier associated with the memory location sent by the first network interface controller to a second device and received by the second network interface controller.

On pages 8 and 9, the Office Action concedes that AAPA “does not show the particular steps recited in the claim for handling identifiers generated by one NIC and received by another NIC in the same machine, in cases when control is transferred [paths changed] between a first network interface and at least a second network interface in a multiple network interface device.” The Office Action then states that Craft, Oesterreicher and Boucher teach limitations of claim 1.

Claim 1 has been amended to recite querying the first network interface controller to supply the program component with *a list of valid identifiers generated* by the first network interface controller, wherein *each identifier from the list of valid identifiers is associated with a location in a memory* of the multiple network interface device (emphasis added). Support for this amendment can be found, for example, on page 15, paragraph 37 of Applicant’s specification. As discussed above, CCB of Craft does not contain such identifiers. Indeed, Craft discusses that “[t]he CCB includes connection information, such as source and destination addresses and ports. For TCP connections a CCB comprises source and destination media access control (MAC) addresses, source and destination Internet Protocol (IP) addresses, source and destination TCP ports and TCP variables such as timers and receive and transmit windows for sliding window protocols.” (Craft, col. 3, line 63 – col. 4, line 2). INIC 22 of Craft is not described to generate a list of valid identifiers..., wherein each identifier from the list of valid identifiers is associated with a location in a memory of the multiple network interface device.

Moreover, Craft does not discuss **querying the first network interface controller to supply** the program component with a list of valid identifiers (emphasis added). Craft discusses that the *ATCP stack maintains* a list of the CCBs that have been offloaded to INICs 22 and 25, and *recognizes that this slow-path packet corresponds to a CCB* that is in a fast-path state (col. 6 lines 47-49) (emphasis added). Thus, in Craft, the ATCP stack “recognizes” the CCB for the fast path. In contrast, claim recites *querying the first network interface controller* to supply the program component with a list of valid identifiers (emphasis added). Craft states that upon receiving this exception condition, the ATCP stack 62 will **command the INIC 25 to flush the fast-path CCB** back to the ATCP stack 62 (col. 6, lines 51-53). Thus, the ATCP stack

determines which CCB to request to be flashed by the INIC 25 and the INIC 25 is not queried “to supply the program component with a list of valid identifiers,” as recited in claim 1.

On page 11, the Office Action states that “it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of AAPA and those of Craft, as discussed above, in order to provide a method of handling received packets when a packet received by one NIC is being handled by other NIC in the same device (col. 6 lines 36-42 in Craft).” However, Craft describes passing **the whole packet** to the INIC device driver which contradicts to a purpose of the RDMA protocol recited in claim 1 (emphasis added). Indeed, the RDMA protocol allows data to move **directly** from the memory of one computer into that of another. Craft discusses that when INIC 22 receives the packet that it cannot process according to the fast-path connection, INIC 22 instead sends **the packet** to the INIC device driver 64 (emphasis added). The ATCP stack of Craft then processes the packet. Thus, a combination of AAPA and Craft would not result in limitations of claim 1 because, while claim 1 recites that the first network interface controller and the second network interface controller operate under a Remote Direct Memory Access (RDMA) protocol and a program component of the multiple network interface device receives the identifier associated with the memory location, Craft discusses sending to the INIC driver the whole packet that INIC 22 cannot process according to the fast-path connection. Sending only an identifier associated with a memory location, which is actually not taught by Craft, would make processing of Craft impossible because Craft is directed to processing of **packets** by INICs. At the same time, sending STag of AAPA along with the packet of Craft will not result in limitations of claim 1 because such combination would no longer describe processing according to the RDMA protocol.

Claim 1 has been amended to recite “determining whether the first network interface controller generated the identifier, wherein when the first network controller generated the identifier the list of valid identifiers comprises the identifier; when it is determined that the first network interface controller generated the identifier, transmitting the memory location associated with the identifier to the second network interface controller, wherein the second network interface controller subsequently transmits the associated data field to the memory location.” Craft does not teach or suggest these limitations.

Furthermore, Craft does not teach or suggest “when the identifier is not found among the list of valid identifiers, invalidating the identifier and discarding the associated data field,” as

recited in claim 1 which has been amended to incorporate subject matter of claim 3. On pages 13 and 14, the Office Action states, with respect to claim 3, that “AAPA in view of Craft, Oesterreicher, Boucher, and in further view of Recio shows that if the identifier has been invalidated, the associated data field is discarded [invalidating STag prevents access to a memory location associated with the identifier] (page 5 lines 15-24; page 12 lines 46-50; page 21 in Recio).” Applicant respectfully notes that Recio does not discuss “invalidating STag” *when the identifier is not found among the list of valid identifiers* (emphasis added). In addition, contrary to the assertions made in the Office Action, none of the cited references teach or suggest when the identifier is not found among the list of valid identifiers, invalidating the identifier and *discarding the associated data field* (emphasis added).

On page 11, the Office Action states that “AAPA in view of Craft and in further view of Oesterreicher does not show that the second network interface controller transmits the associated data field to the memory location.” The Office Action also states that “[as] discussed above, Craft shows that the program component transmits the associated data field to the memory location.” On pages 10 and 22, the Office Action states that Craft shows “transmitting the memory location associated with the identifier to the second network interface controller [commanding the INIC 25 to flush the fast-path CCB back to the ATCP stack for processing the packet, and subsequently handing out the CCB to the INIC 22, which is now associated with the connection] (col. 6 lines 54-57), wherein the [program component] transmits the associated data field to the memory location (col. 6 lines 53-57).” However, if it were true (which is not), as stated in the Office Action, that Craft shows transmitting the memory location associated with the identifier to the second network interface controller while “the [program component] transmits the associated data field to the memory location,” it appears that there would be no need to transmit the memory location associated with the identifier to the second network interface controller. Indeed, the Office Action states that “the [program component]” transmits the associated data field to the memory location; therefore, it appears that transmitting the memory location to the second network interface controller would have been unnecessary.

As discussed above, Craft does not discuss “memory location associated with the identifier” as part of the CCB. Moreover, in the cited passage, Craft states that “[a]fter the packet has been processed by the ATCP stack 62 and the state of the CCB updated to reflect that processing, the CCB can then be handed out to the INIC 22, which is known by port aggregation

driver 66 to be associated with the connection” (col. 6, lines 53-57). Thus, it is the CCB that is handed out to the INIC 22, rather than the associated data field recited in claim 1 is transmitted to the memory location. Claim 1 recites the identifier associated with *the memory location in the multiple network interface device* (emphasis added). Also, claim 1 recites that “the identifier and an associated data field are subsequently received by the second network interface controller in the multiple network interface device from the second device ... the second network interface controller subsequently transmits the associated data field to the memory location.” The CCB of Craft is not received from second device and does not include an associated data field associated with the identifier. It should be clear that by updating the CCB to reflect the processing of the packet by the ATCP stack and handing out the CCB to the INIC 22 Craft does not describe that “the second network interface controller subsequently transmits the associated data field to the memory location.” In addition, Craft does not describe that “the [program component] transmits the associated data field to the memory location” either, as stated in the Office Action. Claim 1

On page 12, the Office Action states that “Boucher shows transmitting the memory location associated with the identifier to the second network interface controller (par. [0018], [0027]-[0029]; Fig. 2), wherein the second network interface controller transmits the associated data field to the memory location (par. [0030]).” Applicant respectfully asserts that Boucher does not teach or suggest **when it is determined that the first network interface controller generated the identifier**, transmitting the memory location associated with the identifier to the second network interface controller (emphasis added).

The Office Action states that “[i]t would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of AAPA in view of Craft and in further view of Oesterreicher by having the second network interface controller transmitting the associated data field to the memory location (instead of the program component, as taught by Craft) in order to offload processing of at least a portion of the packet from the program component to the network interface controller (par. [0008] in Boucher).” However, as discussed above, the cited references do not teach all of the limitations of claim 1. Craft does not mention RDMA. The Office Action states, on page 3 that “RDMA protocol allows for DMA.” However, Craft is not concerned with sending by the first network interface controller an identifier associated with a memory location in the multiple network interface device to a second device, wherein the identifier and an associated data field are subsequently received by the second

network interface controller. Further, as discussed above, Craft does not teach or suggest limitations of claim 1 as asserted in the Office Action. Thus, AAPA does not cure the deficiencies of Craft.

Moreover, not only “AAPA in view of Craft and in further view of Oesterreicher does not show that the second network interface controller transmits the associated data field to the memory location,” as stated in the Office Action, but AAPA in view of Craft and in further view of Oesterreicher do not teach or suggest “when it is determined that the first network interface controller generated the identifier, transmitting the memory location associated with the identifier to the second network interface controller, wherein the second network interface controller subsequently transmits the associated data field to the memory location; and when the identifier is not found among the list of valid identifiers, invalidating the identifier and discarding the associated data field.”

Further, modifying “the teachings of AAPA in view of Craft and in further view of Oesterreicher by having the second network interface controller transmitting the associated data field to the memory location (instead of the program component, as taught by Craft)” does not seem to provide offloading “processing of at least a portion of the packet from the program component to the network interface controller (par. [0008] in Boucher),” as suggested in the Office Action. Indeed, claim 1 recites that “the identifier and an associated data field are subsequently received by the second network interface controller in the multiple network interface device from the second device.” Thus, there is no need to “have” the second network interface controller to transmit the associated data field to the memory location, because the second network interface controller operating under a Remote Direct Memory Access (RDMA) protocol receives the identifier associated with the memory location in the multiple network interface device and the associated data field. Moreover, “the program component, as taught by Craft” does not perform “transmitting the memory location associated with the identifier to the second network interface controller.” In addition, Applicant noted that it seems highly unlikely that one of skill in the art would combine the four cited references to obtain limitations of claim 1.

In view of the foregoing, claim 1 patentably distinguishes over AAPA, Craft, Oesterreicher and Boucher, either alone or in combination.

Claims 2, 4, 5 and 7 depend from claim 1 and are allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 1, 2, 4, 5 and 7 is respectfully requested.

B. Independent Claim 15

Claim 15, as amended, recites:

A computer readable medium having stored therein instructions for performing acts for transferring control between a first network interface controller and at least a second network interface controller in a multiple network interface device, the acts comprising:

after the first network interface controller sends a data request and an identifier associated with a memory location allocated to receive requested data in the multiple network interface device to a second device and the identifier and an associated data field comprising the requested data are subsequently received by the second network interface controller in the multiple network interface device from the second device,

receiving, by a program component in the multiple network interface device, the identifier associated with the memory location in the multiple network interface device and the associated data field from the second device, wherein the second network interface controller has no knowledge of the identifier and the associated data field, and wherein the first network interface controller and the second network interface controller operate under a Remote Direct Memory Access (RDMA) protocol;

querying the first network interface controller to supply the program component with a list of valid identifiers generated by the first network interface controller, wherein each identifier from the list of valid identifiers is associated with a memory location in a memory of the multiple network interface device;

determining whether the first network interface controller generated the identifier, wherein when the first network controller generated the identifier the list of valid identifiers comprises the identifier;

when it is determined that the first network interface controller generated the identifier,

transmitting the memory location associated with the identifier to the second network interface controller, wherein the second network interface controller subsequently transmits the associated data field comprising the requested data to the memory location, and

invalidating the identifier; and

when the identifier is not found among the list of valid identifiers, invalidating the identifier and discarding the associated data field.

On page 12, the Office Action rejects claim 15 for the same reasons as claim 1. As should be clear from the above, AAPA, Craft, Oesterreicher and Boucher do not teach all of the limitations of claim 15.

In addition, claim 15 has been amended to recite after the first network interface controller *sends a data request and an identifier associated with a memory location allocated to receive requested data* in the multiple network interface device to a second device and *the identifier and an associated data field comprising the requested data* are subsequently received by the second network interface controller in the multiple network interface device from the second device (emphasis added). Support for this amendment can be found at least on page 13 of Applicant's specification. The cited references do not teach or suggest this limitation. Further, claim 15 has been amended to recite when it is determined that the first network interface controller generated the identifier, transmitting the memory location associated with the identifier to the second network interface controller, wherein the second network interface controller subsequently transmits the associated data field comprising the requested data to the memory location, and *invalidating the identifier* (emphasis added). Support for this amendment can be found at least on page 15 of Applicant's specification. The cited references do not teach or suggest this limitation either.

In view of the foregoing, claim 15 patentably distinguishes over AAPA, Craft, Oesterreicher and Boucher, either alone or in combination.

Claims 16, 18, 19 and 21 depend from claim 15 and are allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 15, 16, 18, 19 and 21 is respectfully requested.

Rejections Under 35 U.S.C. §103 based on AAPA, Craft, Oesterreicher and Starr

The Office Action rejected claims 8-12, 14, 22-26, and 28 (including independent claims 8 and 22) under 35 U.S.C. §103(a) as allegedly being unpatentable over AAPA in view of Craft, in view of Oesterreicher, in view of Starr et al., US Patent No. 6,807,581 ("Starr"), and in further view of Recio et al., An RDMA Protocol Specification (Version 1.0) ("Recio"). Applicant respectfully disagrees.

C. Independent Claim 8

Claim 8, as amended, recites:

A method for transferring control between a first network interface controller and at least a second network interface controller in a host computer including the first network interface controller and the second network interface controller, the method comprising:

- receiving an identifier and an associated data field in a packet from a remote computer by the second network interface controller, the identifier generated by the first network interface controller and associated with a memory location in the host computer, wherein the second network interface controller has no knowledge of the identifier and the associated data field, and wherein the first network interface controller and the second network interface controller operate under a Remote Direct Memory Access (RDMA) protocol;

- extracting the identifier from the received packet;

- after the identifier has been extracted from the received packet, passing the identifier associated with the memory location to an RDMA program component of the host computer;

- querying, by the RDMA program component, the first network interface controller for a list of valid identifiers generated by the first network interface controller, wherein each identifier from the list of valid identifiers is associated with a memory location in a memory of the host computer;

- searching the list of valid identifiers for the identifier;

- when the list of valid identifiers includes the identifier received from the remote computer, receiving, by the second network interface controller, the memory location associated with the identifier, wherein the second network interface controller transmits the associated data field to the memory location; and

- when the list of valid identifiers does not include the identifier received from the remote computer, invalidating the identifier received from the remote computer and discarding the associated data field.

On page 16, the Office Action concedes that “AAPA does not show the particular steps recited in the claim for handling identifiers generated by one NIC and received by another NIC in the same machine, in cases when control is transferred [paths changed] between a first network interface and at least a second network interface in a multiple network interface device.” The Office Action then states that Craft, Oesterreicher and Boucher teach limitations of claim 8.

The Office Action states that Craft teaches “sending a message [the packet] to a program component indicating the reception of the identifier [INIC 22 sends the packet that it cannot process according to the fast-path connection to the INIC device driver, wherein a program component is interpreted to include at least one or more of the INIC device driver (64), the ATCP stack (62), and the port aggregation driver (66)] (col. 6 lines 43-47; Fig. 1), the program component queries the first network interface controller for a list of identifiers [commanding the

INIC 25 to flush the fast-path CCB back to the ATCP stack, wherein CCB contains a list of identifiers] (col. 3 lines 59 to col. 4 line 9; col. 6 lines 47-53).”

Claim 8 has been amended to recite “receiving an identifier and an associated data field in a packet from a remote computer by the second network interface controller, the identifier generated by the first network interface controller and associated with a memory location in the host computer, wherein the second network interface controller has no knowledge of the identifier and the associated data field, and wherein the first network interface controller and the second network interface controller operate under a Remote Direct Memory Access (RDMA) protocol.” Thus, an identifier associated with a memory location and an associated data field are received by the second network interface controller. As discussed above, Craft does not teach or suggest this limitation.

Furthermore, claim 8 has been amended to recite “extracting the identifier from the received packet; after the identifier has been extracted from the received packet, passing the identifier associated with the memory location to *an RDMA program component* of the host computer” (emphasis added). Support for these amendments can be found at least one pages 3 and 15 of Applicant’s specification. On page 17, the Office Action states that Craft shows “passing the identifier received from the remote computer to the program component [passing the packet that includes identifier in the packet summary to the INIC driver] (col. 4 lines 30-43; col. 6 lines 43-47).” Thus, while claim 8 recites passing the identifier associated with the memory location to *an RDMA program component*, Craft states that the packet is passed *to the INIC driver*. Thus, Craft does not teach Furthermore, Applicant respectfully notes that passing **the whole packet**, as described in Craft, defeats the purpose of utilizing the RDMA protocol recited in claim 1 (emphasis added). Indeed, as well known to one of skill in the art, the RDMA protocol allows data to move **directly** from the memory of one computer into that of another. In addition, as discussed above, claim 8 has been amended to recite *extracting the identifier* from the received packet; *after the identifier has been extracted* from the received packet, passing the identifier associated with the memory location to an RDMA program component of the host computer” (emphasis added). Craft clearly does not teach this limitation claim 8 since, in Craft, the whole packet is passed to the INIC driver.

Further, in the first cited portion, Craft does discuss that upon matching the packet summary with the CCB, assuming no exception conditions exist, the data of the packet, without

network or transport layer headers, *is sent by direct memory access (DMA) units to the destination in storage 23* denoted by the CCB (col. 4, lines 37-41). However, this portion of Craft describes a scenario when the INIC 22 **has** a CCB corresponding to a message received by the INIC 22. Therefore, Craft utilizes DMA in a scenario which is different to the one where the identifier is received by a NIC that is not generated this identifier. Thus, Craft does not discuss “receiving an identifier and an associated data field in a packet from a remote computer by the second network interface controller, the identifier generated by the first network interface controller and associated with a memory location in the host computer,” as recited in claim 8. In another cited passage, Craft discusses a case when the INIC 22 that receives the packet cannot process the packet according to the fast-path connection, and instead **sends the packet** to the INIC device driver 64, which is configured to divert fast-path type message packets to the ATCP stack 62 for processing (col. 6 lines 43-47). Thus, again, the whole packet is sent to the INIC device driver which is different from “after the identifier has been extracted from the received packet, passing the identifier associated with the memory location,” as recited in claim 8.

In view of the above, a combination of AAPA and Craft is improper because Craft describes passing the whole packet to the INIC, AAPA states that NIC 2 receives Stag generated by NIC 1 and claim 8 recites “passing the identifier associated with the memory location to a program component of the host computer.”

The Office Action alleges that Starr “shows: searching the list of identifiers [comparing packet summary with CCB hashes and CCB cache] (Fig. 3 step (110); col. 9 lines 40-44); when the list of identifiers includes the identifier received from the remote computer, receiving a memory location associated with the identifier [if the packet summary matches a CCB, receiving a memory location according to a file system] (Fig. 3 steps (120) and (122); col. 9 lines 55-61); and when the list of identifiers does not include the identifier received from the remote computer [if the packet summary does not match a CCB] (Fig. 3 step (110); col. 9 lines 44-46, [sending packet to stack for slow-path processing] (Fig. 3 step (112); col. 9 lines 44-46).”

Claim 8 as amended recites querying, by *the RDMA program component*, the first network interface controller for a list of valid identifiers generated by the first network interface controller, *wherein each identifier from the list of valid identifiers is associated with a memory location in a memory of the host computer* (emphasis added). Starr does not teach or suggest this limitation. The CCB of Starr does not include a list of valid identifiers each associated with a

memory location generated by the first network interface controller. Further, Starr does not teach or suggest querying, by *the RDMA program component*, the first network interface controller ... (emphasis added). Moreover, none of the other cited references teaches or suggests this limitation.

Claim 8 has been amended to incorporate portion of subject matter of claim 10 and to thus recite “when the list of valid identifiers does not include the identifier received from the remote computer, invalidating the identifier received from the remote computer and discarding the associated data field.” On page 20, the Office Action states, with respect to claim 10, that “AAPA in view of Craft, Oesterreicher, Starr, and in further view of Recio shows that if the identifier has been invalidated, the associated data field is discarded [invalidating STag prevents access to a memory location associated with the identifier] (page 5 lines 15-24; page 12 lines 46-50; page 21 in Recio).” Applicant respectively notes that Recio does not discuss “invalidating STag” *when the identifier is not found among the list of valid identifiers* (emphasis added). In addition, contrary to the assertions made in the Office Action, none of the cited references teach or suggest when the list of valid identifiers does not include the identifier received from the remote computer, invalidating the identifier and *discarding the associated data field* (emphasis added).

In view of the foregoing, claim 8 patentably distinguishes over AAPA, Craft, Starr and Recio, either alone or in combination.

Claims 9, 11, 12 and 14 depend from claim 8 and are allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 8, 9, 11, 12 and 14 is respectfully requested.

D. Independent Claim 22

Claim 22, as amended, recites:

A computer readable medium having stored therein instructions for performing acts for transferring control between a first network interface controller and at least a second network interface controller in a host computer including the first network interface controller and the second network interface controller, the acts comprising:

receiving an identifier and an associated data field in a packet from a remote computer by the at least a second network interface controller, the identifier generated by the first network interface controller and associated with a memory location in the host computer, wherein the second network interface controller has no knowledge of the identifier and the associated data field, and wherein the first network interface controller and the second network interface controller operate under a Remote Direct Memory Access (RDMA) protocol;

extracting the identifier from the received packet;

after the identifier has been extracted from the received packet, passing the identifier associated with the memory location to a program component of the host computer;

querying, by the program component, the first network interface controller for a list of valid identifiers generated by the first network interface controller, wherein each identifier from the list of valid identifiers is associated with a memory location in a memory of the host computer;

searching the list of valid identifiers for the identifier;

when the list of valid identifiers includes the identifier received from the remote computer, receiving, by the second network interface controller, the memory location associated with the identifier, wherein the second network interface controller transmits the associated data field to the memory location; and

when the list of valid identifiers does not include the identifier received from the remote computer, invalidating the identifier received from the remote computer and discarding the associated data field.

The Office Action appears to reject claim 22 for the same reasons as claim 8. As should be clear from the above, even if the cited references were combined, the combination would not teach or suggest all limitations of claim 22. In particular, the cited references do not teach or suggest “extracting the identifier from the received packet; after the identifier has been extracted from the received packet, passing the identifier associated with the memory location to a program component of the host computer,” as recited in claim 22. Further, the cited references do not teach or suggest “querying, by the program component, the first network interface controller for a list of valid identifiers generated by the first network interface controller, wherein each identifier from the list of valid identifiers is associated with a memory location in a memory of the host computer; searching the list of valid identifiers for the identifier; when the list of valid

identifiers includes the identifier received from the remote computer, receiving, by the second network interface controller, the memory location associated with the identifier, wherein the second network interface controller transmits the associated data field to the memory location; and when the list of valid identifiers does not include the identifier received from the remote computer, invalidating the identifier received from the remote computer and discarding the associated data field,” as also recited in claim 22.

In view of the foregoing, claim 22 patentably distinguishes over AAPA, Craft, Starr and Recio, either alone or in combination.

Claims 23, 25, 26 and 28 depend from claim 22 and are allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 22, 23, 25, 26 and 28 is respectfully requested.

CONCLUSION

A Notice of Allowance is respectfully requested. The Examiner is requested to call the undersigned at the telephone number listed below if this communication does not place the case in condition for allowance.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicant hereby requests any necessary extension of time. If there is a fee occasioned by this response, including an extension fee, the Director is hereby authorized to charge any deficiency or credit any overpayment in the fees filed, asserted to be filed or which should have been filed herewith to our Deposit Account No. 23/2825, under Docket No. M1103.70194US00.

Dated: April 14, 2009

Respectfully submitted,

By /James H. Morris/
James H. Morris
Registration No.: 34,681
WOLF, GREENFIELD & SACKS, P.C.
Federal Reserve Plaza
600 Atlantic Avenue
Boston, Massachusetts 02210-2206
617.646.8000